ELSEVIER

Discrete Applied Mathematics 94 (1999) 297–319

DISCRETE APPLIED MATHEMATICS

A technique to find multiple motif occurrences in a biomolecular sequence

Nicolas Moeri[a,b,*]

[a] *Département de Mathématiques, Ecole Polytechnique Fédérale, MA Ecublens, CH-1015 Lausanne, Switzerland*

[b] *Bioinformatics Group, Swiss Institute for Experimental Cancer Research, CH-1066 Epalinges, Switzerland*

Received 24 December 1996; revised 2 July 1997; accepted 8 May 1998

Abstract

A new flexible and efficient search technique has been recently introduced which includes two main components, a generalized profile syntax serving as motif definition language, and a motif search technique finding multiple instances of a motif in the same sequence. The generalized profile structure is presented in detail, as well as the *alignment* and *score* notions which constitute the foundations of modern motif descriptors. In order to mathematically model these notions, given a sequence and a profile, an *alignment graph* is built, so that a motif occurrence in the sequence is equivalent to a path in this directed graph. A mathematical statement of the motif search problem is formulated based on "reasonable" biological considerations. This statement together with an alignment *disjointness* definition allows finally to implement an efficient algorithm solving the stated problem exactly.

1. Introduction

The ultimate goal of most genome sequencing programs is to understand the information contained in a genetic program. After having defined the complete base sequence of an organism's genome or the complete amino acid sequence of its protein inventory, the main challenge lies in the interpretation of these data by automatic procedures. Understanding the information contained in these data consists of predicting the underlying biological function by applying some well-defined rules on a base sequence. One major difficulty in this process resides in the degeneracy of genetic coding mechanisms. Therefore, gene expression signals having the same function, or protein domains having similar 3D-structures can exhibit a high degree of sequence variations.

* Correspondence address: Département de Mathématiques, Ecole Polytechnique Fédérale, MA Ecublens, CH-1015 Lausanne, Switzerland. E-mail: moeri@dma.epfl.ch.

0166-218X/99/$ see front matter
PII: S0166-218X(99)00027-X

Despite these variations, these sequences share some common properties. The synthesis of these common properties is called a *sequence motif*.

The role of a motif search technique is to decompose a large sequence, describing a gene or a protein, into smaller subsequences that are expressions of known sequence motifs. In a typical application, a new sequence is compared against a library of known motifs. To implement such an application, a motif search technique has two distinct interdependent components, a *motif descriptor* and a search method to locate motif instances in a particular sequence.

In this paper, we review a new motif descriptor called a *generalized profile* [5] which can emulate most of other used motif descriptors. In order to define the role of the generalized profiles, we introduce the notions of *alignment* and *alignment score* in the general context of comparison between some motif descriptor and a sequence. We finally apply these notions to the generalized profile itself.

Based on this new motif descriptor, one goal of the paper is to define a search method by an exact formulation of the mathematical problem, leaving no ambiguities to its implementation. A second goal of the paper consists of describing a new efficient algorithm solving the stated problem. We show that this algorithm produces an adequate solution to the motif search problem, and some results compared to a naive approach of the problem witness the algorithm's efficiency.

2. Basic data definitions of the search problem

A standard motif search problem involves two data components:

- a biomolecular *sequence*;
- a *motif descriptor*.

We consider a biomolecular *sequence* as a string of symbols from an *alphabet*.

Definition 1. An *alphabet* Λ is defined by a set of symbols $\{\lambda_1, \ldots, \lambda_t\}$.

For example, DNA sequences are usually described by symbols from the nucleotides alphabet $\{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$, and proteins by symbols from the amino acids alphabet $\{\mathtt{A}, \mathtt{C}, \mathtt{D}, \mathtt{E}, \mathtt{F}, \mathtt{G}, \mathtt{H}, \mathtt{I}, \mathtt{K}, \mathtt{L}, \mathtt{M}, \mathtt{N}, \mathtt{P}, \mathtt{Q}, \mathtt{R}, \mathtt{S}, \mathtt{T}, \mathtt{V}, \mathtt{W}, \mathtt{Y}\}$.

Definition 2. Given an alphabet Λ, a *sequence* based on Λ is a word on Λ.

Definition 3. Given an alphabet Λ, a *motif* $\boldsymbol{S}$ based on Λ is a language on Λ. If a sequence belongs to $\boldsymbol{S}$, the sequence is called a *motif instance*.

Locating a motif occurrence in a sequence consists in identifying a *subsequence* as a motif instance.

Definition 4. Given a sequence a of length n and two indices s and t such that $1 \leqslant s \leqslant t \leqslant n$, let $a_{s,t}$ denote the *subsequence* of a defined by the sequence $a_s \ldots a_t$.

A motif is described by a data structure called a *motif descriptor*. A motif descriptor allows to characterize a subsequence in relation to a motif. Modern motif descriptors such as hidden Markov models [10] or generalized profiles are based on the *alignment* and *score* concepts. Such motif descriptors are defined by a *size*, which is a positive integer value, and an *alignment score function*. In what follows, we consider only this kind of motif descriptors.

2.1. The alignment concept

The alignment notion is the key of modern motif descriptors as it takes into account any possible degeneracy or sequencing mistakes occurring in a biomolecular sequence. Given a motif descriptor of size m, we define a set of m indexed slots $\{s_1, \ldots, s_m\}$. We introduce the notion of *alignment* between the motif descriptor and a sequence as the way of mapping a set of symbols of the sequence to these slots through a list of *alignment operations*.

Definition 5. Given a motif descriptor of size m and a sequence $a = a_1 \ldots a_n$, the possible *alignment operations* between the motif descriptor and the sequence are

- the match operation $\mathscr{M}(x, y)$ with $1 \leqslant x \leqslant m$ and $1 \leqslant y \leqslant n$; symbol a_y is matched with slot s_x;
- the insert operation $\mathscr{I}(x, y)$ with $0 \leqslant x \leqslant m$ and $1 \leqslant y \leqslant n$; symbol a_y is inserted after slot s_x if $x > 0$ and before slot s_{x+1} if $x < m$;
- the delete operation $\mathscr{D}(x, y)$ with $1 \leqslant x \leqslant m$ and $0 \leqslant y \leqslant n$; the slot s_x is deleted; index y only makes sense in the context of an alignment where it indicates that only symbols a_y and a_{y+1} can be involved in the previous and the next alignment operation, respectively.

Definition 6. Given a motif descriptor and a sequence, an *alignment* A of length l between the motif descriptor and the sequence is defined by a list of l alignment operations such that each pair of consecutive alignment operations $\mathscr{O}_1(x_1, y_1)$ and $\mathscr{O}_2(x_2, y_2)$ respects the following conditions:

- $\mathscr{O}_2 = \mathscr{M} \Rightarrow (x_2, y_2) = (x_1, y_1) + (1, 1)$,
- $\mathscr{O}_2 = \mathscr{I} \Rightarrow (x_2, y_2) = (x_1, y_1) + (0, 1)$,
- $\mathscr{O}_2 = \mathscr{D} \Rightarrow (x_2, y_2) = (x_1, y_1) + (1, 0)$.

Given an alignment between a sequence and a motif descriptor, the set of symbols involved in an alignment operation constitutes the subsequence *mapped* to the motif descriptor. For example, given a motif descriptor of size 6 and the sequence

$a = \texttt{TAGATT}$, a sample of an alignment is given by

$$A = \{\mathcal{M}(2,2), \mathcal{D}(3,2), \mathcal{M}(4,3), \mathcal{I}(4,4), \mathcal{I}(4,5), \mathcal{M}(5,6)\}. \tag{1}$$

Subsequence $a_{2,6} = \texttt{AGATT}$ is mapped to the motif descriptor by this alignment through the operations $\mathcal{M}(2,2), \mathcal{M}(4,3), \mathcal{I}(4,4), \mathcal{I}(4,5), \mathcal{M}(5,6)$. When the mapped subsequence is known, an alignment can be described by the usual alignment matrix presented in [15]. In this case, this matrix would be

$$\begin{bmatrix} s_2 & s_3 & s_4 & - & - & s_5 \\ \texttt{A} & - & \texttt{G} & \texttt{A} & \texttt{T} & \texttt{T} \end{bmatrix}. \tag{2}$$

2.2. The alignment graph

In order to visualize an alignment, we can build a graph where each edge corresponds to an alignment operation and each path corresponds to an alignment.

Definition 7. Given a motif descriptor M of size m and a sequence a of size n, the *alignment graph* $G(M,a)$ is defined by the vertices set

$$\{\mathcal{V}_{x,y} \text{ for } x = 0,\ldots,m \text{ and } y = 0,\ldots,n\} \tag{3}$$

and the edges set

$$\left\{ \begin{array}{lll} \mathcal{V}_{x,y}\mathcal{V}_{x+1,y+1} & \text{for } x = 0,\ldots,m-1 & \text{and } y = 0,\ldots,n-1 \\ \mathcal{V}_{x,y}\mathcal{V}_{x,y+1} & \text{for } x = 0,\ldots,m & \text{and } y = 0,\ldots,n-1 \\ \mathcal{V}_{x,y}\mathcal{V}_{x+1,y} & \text{for } x = 0,\ldots,m-1 & \text{and } y = 0,\ldots n \end{array} \right\}. \tag{4}$$

Definition 8. Given a non-empty alignment A between a motif descriptor M and a sequence a, the corresponding alignment path $P(A)$ in $G(M,a)$ is uniquely defined by the following relationships:

$$\mathcal{M}(x,y) \in A \Leftrightarrow \mathcal{V}_{x-1,y-1}\mathcal{V}_{x,y} \in P(A), \tag{5}$$

$$\mathcal{I}(x,y) \in A \Leftrightarrow \mathcal{V}_{x,y-1}\mathcal{V}_{x,y} \in P(A), \tag{6}$$

$$\mathcal{D}(x,y) \in A \Leftrightarrow \mathcal{V}_{x-1,y}\mathcal{V}_{x,y} \in P(A). \tag{7}$$

Property 9. *The relationships expressed in Definition* 8 *define a bijection between non-empty alignments and alignment paths.*

Fig. 1 shows the alignment graph and the alignment path corresponding to the alignment shown in Eq. (1). The alignment path is defined by

$$P = \{\mathcal{V}_{1,1}\mathcal{V}_{2,2}; \mathcal{V}_{2,2}\mathcal{V}_{3,2}; \mathcal{V}_{3,2}\mathcal{V}_{4,3}; \mathcal{V}_{4,3}\mathcal{V}_{4,4}; \mathcal{V}_{4,4}\mathcal{V}_{4,5}; \mathcal{V}_{4,5}\mathcal{V}_{5,6}\}. \tag{8}$$

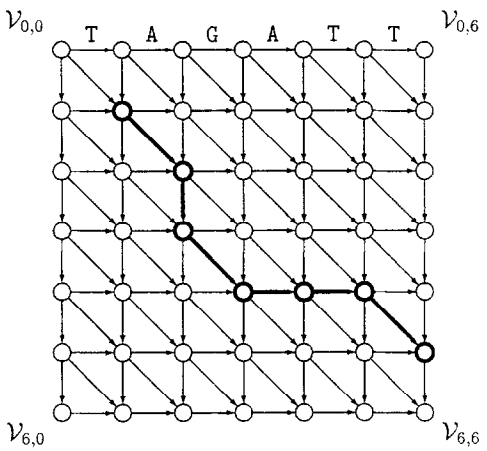

Fig. 1. Sample of alignment graph with an alignment path shown in bold.

2.3. The alignment score

Given a motif descriptor and a sequence, the main goal of an alignment is to determine whether the sequence contains some subsequences similar to the described motif. This similarity is quantified by the *alignment score*.

Definition 10. Given a motif descriptor and a sequence, an *alignment score* is defined by a function $S : A \rightarrow R$ where A is the set of all possible alignments between the motif descriptor and the sequence.

As several alignments can map the same subsequence to the motif descriptor, we usually consider an alignment obtaining the highest score. For various biological reasons, we are often not only interested in this alignment, but also in a set of alignments obtaining high scores, each alignment implying a different biological hypothesis (see Fig. 11).

In order to be considered as a motif instance, a subsequence must be mapped by an alignment whose score is greater than or equal to a real value called the *cut-off value*, which guarantees that the subsequence has a minimal degree of similarity with the motif. The motif search problem can therefore be expressed as an *alignment search problem*, where each alignment defines not only a subsequence, but also a similarity between the subsequence and the motif, and the way the motif has been conserved in the subsequence.

2.4. The Smith–Waterman scoring function

The first local alignment scoring function has been introduced in [16]. This scoring function was designed to find similarities between two sequences. We show in this

section that a two sequences comparison can be expressed as a motif search problem using the formalism previously described. Given two sequences, we consider one as the reference sequence, and the other as the sequence to test. A motif descriptor of size m includes the following components:

- a reference sequence of size m based on alphabet Λ;
- a substitution matrix $B = [b_{\mu,\nu}] \in M_{|\Lambda|\times|\Lambda|}$ with $\mu, \nu \in \Lambda$;
- a gap weighting function $w : \boldsymbol{N}^+ \to \boldsymbol{R}$ defined by $w(k) = \alpha k + \beta$ with $\alpha, \beta \in \boldsymbol{R}$.

We suppose that the sequence to test is based on Λ. The substitution matrix gives a cost to the match operations, and the *gap* weighting function penalizes a succession of insert or delete operations.

Definition 11. Given an alignment A, a *gap* is a maximal ordered non-empty subset of A containing only insert operations or only delete operations consecutive in A. The length of gap γ is the size $|\gamma|$ of the subset. The set of all gaps of A is denoted by $\Gamma(A)$.

Definition 12. Given a motif descriptor defined by the components enumerated above and a sequence based on Λ, the score of A is given by

$$S(A) = \sum_{\mathscr{M}(x,y)\in A} B_{a_x,b_y} - \sum_{\gamma\in\Gamma(A)} w(|\gamma|). \tag{9}$$

Note 1. *This alignment scoring method is independent of the mapped subsequences' positions in the sequences.*

Consider the reference sequence GAAGTA of size 6 based on alphabet $\{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}$, and the alignment A presented in Eq. (1). The gaps set of A is $\Gamma(A) = \{\{\mathscr{D}(3,2)\}, \{\mathscr{I}(4,4),\mathscr{I}(4,5)\}\}$. Given the substitution matrix

$$B = \begin{array}{c} \\ \mathtt{A} \\ \mathtt{C} \\ \mathtt{G} \\ \mathtt{T} \end{array} \begin{array}{c} \begin{array}{cccc} \mathtt{A} & \mathtt{C} & \mathtt{G} & \mathtt{T} \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 \end{pmatrix} \end{array} \tag{10}$$

and the gap weighting function w defined by $\alpha = \beta = 1$, the alignment score of A is

$$S(A) = B_{\mathtt{A},\mathtt{A}} + B_{\mathtt{G},\mathtt{G}} + B_{\mathtt{T},\mathtt{T}} - (w(1) + w(2)) = 1 + 3 + 4 - (2 + 3) = 3. \tag{11}$$

The most current application using this scoring function is the search of an alignment between two given sequences maximizing its score. The Smith–Waterman algorithm computes the score of this optimal alignment which often constitutes the first value we are interested in. An efficient implementation of this algorithm is presented in [6].

3. The generalized profile

Generalized profiles are an extension of the profiles introduced in [8,7] including some additional parameters allowing in particular a more flexible treatment of the insertions and deletions, and an emulation of different alignment modes [5]. Generalized profiles are based on the two concepts previously described: *alignment* and *scoring function*.

As sequences, generalized profiles have a linear structure where symbols are replaced by *match positions*, each match position is surrounded by two *insert positions*. The number of match positions defines the size of the generalized profile. The match positions and the insert positions contain all the parameters defining the alignment score function.

3.1. The generalized profile parameters

A profile M of size m based on Λ is defined by a set of parameters contained in m match positions $\mathrm{MP}^1,\ldots,\mathrm{MP}^m$ and $m+1$ insert positions $\mathrm{IP}^0,\ldots,\mathrm{IP}^m$. We define two operations in addition to the alignment operations previously described; the begin operation $\mathscr{B}(x,y)$ and the end operation $\mathscr{E}(x,y)$ which occur respectively before starting and after ending an alignment. Given a non-empty alignment $A=\{\mathscr{C}_1(x_1,y_1),\ldots,\mathscr{C}_l(x_l,y_l)\}$, we define the begin and the end operations of A by

$$\mathscr{B}(A)=\begin{cases}\mathscr{B}(x_1-1,y_1-1) & \text{if } \mathscr{C}_1=\mathscr{M},\\ \mathscr{B}(x_1-1,y_1) & \text{if } \mathscr{C}_1=\mathscr{I} \text{ and } \mathscr{E}(A)=\mathscr{E}(x_l,y_l),\\ \mathscr{B}(x_1,y_1-1) & \text{if } \mathscr{C}_1=\mathscr{D}.\end{cases} \tag{12}$$

Definition 13. Given a non-empty alignment $A=\{\Omega_1,\ldots,\Omega_l\}$, we define the *extended alignment* $\bar{A}$ by $\bar{A}=\{\mathscr{B}(A),\Omega_1,\ldots,\Omega_l,\mathscr{E}(A)\}$.

The main function of the generalized profile is to score each operation and each transition between each possible consecutive operations pair of an extended alignment.

Given a profile based on Λ, the match position MP^x contains the following parameters:

- the match parameters $m^x=[m^x_\lambda]\in\boldsymbol{R}^{|\Lambda|}$ with $\lambda\in\Lambda$;
- the deletion parameter $d^x\in\boldsymbol{R}$.

The insert position IP^x contains the following parameters:

- the insert parameters $i^x=[i^x_\lambda]\in\boldsymbol{R}^{|\Lambda|}$ with $\lambda\in\Lambda$;
- the begin parameters $\hat{b}^x,\tilde{b}^x\in\boldsymbol{R}$;
- the end parameters $\hat{e}^x,\tilde{e}^x\in\boldsymbol{R}$;
- the transition parameters $t^x_{\mathscr{C}_1,\mathscr{C}_2}\in M_{4\times4}$ with $(\mathscr{C}_1,\mathscr{C}_2)\in\{\mathscr{B},\mathscr{M},\mathscr{I},\mathscr{D}\}\times\{\mathscr{M},\mathscr{I},\mathscr{D},\mathscr{E}\}$.

Table 1
Summary of the generalized parameters

Ω	$S_{\text{op}}(\Omega)$
$\mathscr{M}(x, y)$	$m^x_{a_y}$
$\mathscr{D}(x, y)$	d^x
$\mathscr{I}(x, y)$	$i^x_{a_y}$
$\mathscr{B}(x, y = 0)$	$\hat{b}^x$
$\mathscr{B}(x, y > 0)$	$\tilde{b}^x$
$\mathscr{E}(x, y = n)$	$\hat{e}^x$
$\mathscr{E}(x, y < n)$	$\tilde{e}^x$
(Ω_1, Ω_2)	$S_{\text{trans}}(\Omega_1, \Omega_2)$
$(\mathscr{C}_1(x_1, y_1), \mathscr{C}_2(x_2, y_2))$ with $(\mathscr{C}_1, \mathscr{C}_2) \in \{\mathscr{B}, \mathscr{M}, \mathscr{I}, \mathscr{D}\} \times \{\mathscr{M}, \mathscr{I}, \mathscr{D}, \mathscr{E}\}$	$t^{x_1}_{\mathscr{C}_1, \mathscr{C}_2}$

Table 1 summarizes the parameters contained in a generalized profile, and defines the functions S_{op} and S_{trans} scoring respectively the alignment operations and the possible transitions between consecutive alignment operations in an alignment.

3.2. The profile alignment score

The scoring functions S_{op} and S_{trans} defined by the profile allows us to introduce the alignment score function.

Definition 14. Given a non-empty alignment A between a profile and a sequence, we consider the corresponding extended alignment $\tilde{A} = \{\Omega_1, \ldots, \Omega_q\}$. The *score* of A is defined by

$$S(A) = \sum_{k=1}^{q} S_{\text{op}}(\Omega_k) + \sum_{k=2}^{q} S_{\text{trans}}(\Omega_{k-1}, \Omega_k). \tag{13}$$

3.3. The profile alignment graph

To solve any profile alignment problem, it is useful to formulate the problem in terms of directed graphs. Given a profile M of size m and a test sequence a of size n, we build the *profile alignment graph* $G(M, a)$ based on the basic alignment graph presented in Section 2.2.

In order to introduce all the parameters contained in a generalized profile, each vertex $\mathscr{V}_{x,y}$ of the basic alignment graph is replaced by a *node* $\mathscr{N}_{x,y}$ to include the transition scores. A node consists of a perfect matching between vertices $\mathscr{V}_{x,y}$ with $\mathscr{V} = \mathscr{B}, \mathscr{M}, \mathscr{I}, \mathscr{D}$ and vertices $\mathscr{W}^+_{x,y}$ with $\mathscr{W} = \mathscr{M}, \mathscr{I}, \mathscr{D}, \mathscr{E}$. Edge $\mathscr{V}_{x,y}\mathscr{W}_{x,y}$ corresponds to a transition from an operation $\mathscr{V}$ to an operation $\mathscr{W}$. The nodes are interconnected as follows:

Table 2
Costs assigned to the edges of the alignment graph

Edge	Length
$\mathscr{M}^+_{x-1,y-1}\mathscr{M}_{x,y}$	$m^x_{a_y}$
$\mathscr{D}^+_{x-1,y}\mathscr{D}_{x,y}$	d^x
$\mathscr{I}^+_{x,y-1}\mathscr{I}_{x,y}$	$i^x_{a_y}$
$\mathscr{B}^+\mathscr{B}_{x,y}$	$b^x(y)$
$\mathscr{E}^+_{x,y}\mathscr{E}$	$e^x(y)$
$\mathscr{T}_{x,y}\mathscr{U}_{x,y}$	$t^x_{a_y,a_{y'}}$
with $(\mathscr{T},\mathscr{U}) \in \{\mathscr{B},\mathscr{M},\mathscr{I},\mathscr{D}\} \times \{\mathscr{M},\mathscr{I},\mathscr{D},\mathscr{E}\}$	

- nodes $\mathscr{N}_{x,y}$ and $\mathscr{N}_{x+1,y+1}$ are connected by edge $\mathscr{M}^+_{x,y}\mathscr{M}_{x+1,y+1}$ corresponding to operation $\mathscr{M}(x,y)$ for $x=0,\ldots,m-1$ and $y=0,\ldots,n-1$;
- nodes $\mathscr{N}_{x,y}$ and $\mathscr{N}_{x,y+1}$ are connected by edge $\mathscr{I}^+_{x,y}\mathscr{I}_{x,y+1}$ corresponding to operation $\mathscr{I}(x,y+1)$ for $x=0,\ldots,m$ and $y=0,\ldots,n-1$;
- nodes $\mathscr{N}_{x,y}$ and $\mathscr{N}_{x-1,y}$ are connected by edge $\mathscr{D}^+_{x,y}\mathscr{D}_{x+1,y}$ corresponding to operation $\mathscr{D}(x+1,y)$ for $x=0,\ldots,m-1$ and $y=0,\ldots,n$.

To introduce the begin and the end scores, we add a source vertex $\mathscr{B}^+$ and a destination vertex $\mathscr{E}$. Each node $\mathscr{N}_{x,y}$ is connected to vertices $\mathscr{B}^+$ and $\mathscr{E}$ by edges $\mathscr{B}^+\mathscr{B}_{x,y}$ and $\mathscr{E}^+_{x,y}\mathscr{E}$ corresponding respectively to operations $\mathscr{B}(x,y)$ and $\mathscr{E}(x,y)$.

Each edge corresponds to a possible operation between profile M and the test sequence a or to an operations transition, so we assign to each edge a length equal to the score of the corresponding operation or transition as shown in Table 2 using the following definitions:

$$b^x(y)=\begin{cases}\hat{b}^x & \text{if } y=0,\\ \tilde{b}^x & \text{otherwise,}\end{cases} \quad \text{and} \quad e^x(y)=\begin{cases}\hat{e}^x & \text{if } y=n,\\ \tilde{e}^x & \text{otherwise.}\end{cases} \tag{14}$$

Definition 15. Given a non-empty alignment A between a profile and a sequence, the *profile alignment path* $P(A)$ in the profile alignment graph is entirely defined by the following relationships:

$$\mathscr{B}(x,y)\in\bar{A} \Leftrightarrow \mathscr{B}^+\mathscr{B}_{x,y}\in P(A), \tag{15}$$

$$\mathscr{M}(x,y)\in\bar{A} \Leftrightarrow \mathscr{M}^+_{x-1,y-1}\mathscr{M}_{x,y}\in P(A), \tag{16}$$

$$\mathscr{I}(x,y)\in\bar{A} \Leftrightarrow \mathscr{I}^+_{x,y-1}\mathscr{I}_{x,y}\in P(A), \tag{17}$$

$$\mathscr{D}(x,y)\in\bar{A} \Leftrightarrow \mathscr{D}^+_{x-1,y}\mathscr{D}_{x,y}\in P(A), \tag{18}$$

$$\mathscr{E}(x,y)\in\bar{A} \Leftrightarrow \mathscr{E}^+_{x,y}\mathscr{E}\in P(A). \tag{19}$$

Property 16. *The relationships expressed in Definition* 15 *define a bijection between non empty alignments and paths from vertex* $\mathscr{B}^+$ *to vertex* $\mathscr{E}$ *not containing an edged* $\mathscr{B}_{x,y}\mathscr{E}^+_{x,y}$ *in the profile alignment graph. Such paths are called* alignment paths.

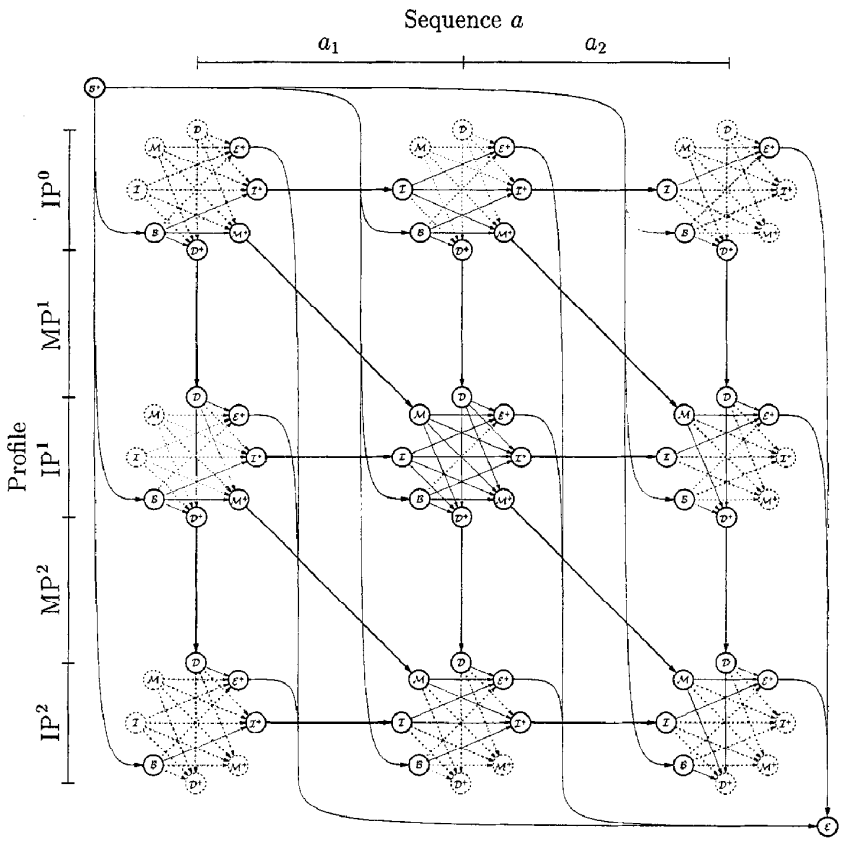

Fig. 2. Sample of alignment graph. Vertex indices are not shown.

Property 17. *Let* $L : \boldsymbol{P} \rightarrow \boldsymbol{R}$, *where* $\boldsymbol{P}$ *is the set of all possible alignment paths in* $G(M,a)$, *denotes the function defining the length of an alignment path. Then, for any non-empty alignment* A, $S(A) = L(P(A))$.

Fig. 2 shows the alignment graph for a profile and a test sequence of lengths 2. As any alignment path starts from vertex $\mathscr{B}^+$ and ends at vertex $\mathscr{E}$ and does not contain a $\mathscr{B}_{x,y}\mathscr{E}^+_{x,y}$ edge, dashed vertices and dashed edges are never used.

3.3.1. Computation of the optimal alignment score

Given a profile M of size m and a sequence a of size n, the score of a best alignment between M and a is given by the length of a longest path from the source vertex $\mathscr{B}^+$ to the destination vertex $\mathscr{E}$ in the alignment graph $G(M,a)$.

Let $\hat{L}_{\mathscr{B}}(\mathscr{V})$ denote the length of a longest path from vertex $\mathscr{B}^+$ to vertex $\mathscr{V}$. Knowing $\hat{L}_{\mathscr{B}}(\mathscr{E}^+_{x,y})$ for $0 \leqslant x \leqslant m$ and $0 \leqslant y \leqslant n$ allows to compute $\hat{L}_{\mathscr{B}}(\mathscr{E})$. Due to the regular structure of the alignment graph, in the general case $x > 0$ and $y > 0$, we can compute

$\hat{L}_{\mathscr{M}}(\mathscr{V}^{+}_{x,y})$ for $\mathscr{V} \in \{\mathscr{M}, \mathscr{I}, \mathscr{D}, \mathscr{E}\}$ when $\hat{L}_{\mathscr{M}}(\mathscr{V}^{+}_{x-1,y-1})$, $\hat{L}_{\mathscr{M}}(\mathscr{V}^{\downarrow}_{x-1,y})$ and $\hat{L}_{\mathscr{M}}(\mathscr{V}^{\rightarrow}_{x,y-1})$ are known for $\mathscr{V} \in \{\mathscr{M}, \mathscr{I}, \mathscr{D}, \mathscr{E}\}$. The resulting algorithm complexity is $O(mn)$, and the memory storage $O(m)$. A detailed description of these algorithm is given in [5].

4. The motif search problem

The goal of a motif search method consists of finding meaningful instances of motifs in a given collection of sequences. In a typical application, one is interested in the following questions:

(1) Which sequences do contain the motif ?
(2) How many times does the motif occur in the sequence?
(3) Where are the motif instances located?
(4) How similar are these motif instances to the motif?
(5) How can the motif instances be aligned to the motif?

It is important to recognize that these questions cannot be fully answered by formulating the motif search problem as a classification problem, which answers only the question (1). Indeed, published database search algorithms using profiles or other motif descriptor models are designed mainly to find only the single best alignment between the model and a sequence, or to compute only a single value to assess membership of a sequence family. The advantage of the motif search method defined here is that its solution provides a more satisfactory answer to the above questions.

There are two reasons why the search for a biomolecular sequence motif is not a trivial task, even if the motif is well defined by an appropriate descriptor. The first one is that genetic texts, in the form of nucleotide sequences or translated into protein, do not contain any obvious punctuation signals. As a consequence, delineation of functional subsequences and classification of these subsequences must proceed simultaneously. The second reason is that biological sequence motifs, of the same or of different types, can occur in partially overlapping way. The high degeneracy of many motifs favors such an arrangement. However, the physical overlap constraints vary greatly between different motifs. Therefore, a generally usable motif search technique must deal with the overlap problem in a flexible way.

4.1. Motif search problem for generalized profiles

Given a generalized profile and a biomolecular sequence, the result of a motif search operation has the form of a set of alignments whose corresponding similarity scores must exceed a given cut-off value. As a first approximation, the goal can be described as finding all alignments with scores higher than the prescribed cut-off value, but practically, the size of such an alignment set grows exponentially with the size of the profile and the length of the sequence. For example, each alignment exceeding the cut-off value is usually surrounded by a large number of similar alignments also

exceeding the cut-off value. Usually we want such a group of alignments to be represented by a single, locally optimal alignment. This can be achieved by requiring that two alignments contained in the result of a motif search operation satisfy a specific disjointness relationship. By extension, this disjointness relationship can be applied to alignment paths. A formal statement of the motif search problem can be described as follows.

Given a sequence a, a profile M, a symmetric disjointness relationship $\|$ between two alignments and a cut-off value α, the problem consists in finding a (not necessarily unique) set of alignments between M and a, or equivalently a set $\boldsymbol{E}$ of feasible paths in the alignment graph $G(M,a)$ satisfying conditions (1)–(4) below.

(1) *Minimal length condition.* The length of each alignment path of the set is greater than or equal to the cut-off value: $\forall P \in \boldsymbol{E}$, $L(P) \geqslant \alpha$.

(2) *Disjointness condition.* Any two paths in the set are disjoint: $\forall P, P' \in \boldsymbol{E}$ with $P \neq P'$, $P \| P'$.

(3) *Local maximality condition.* Given any alignment path in $G(M,a)$ whose length exceeds the cut-off value, there exists an alignment path in the solution set not disjoint from the given path with an equal or a greater length: $\forall$ alignment path $P' \in G(M,a)$ with $L(P') \geqslant \alpha$, $\exists P \in \boldsymbol{E}$ such that $P \nparallel P'$ and $L(P) > L(P')$.

(4) *Set maximality condition.* No profile alignment path in $G(M,a)$ whose length is greater than or equal to the cut-off value can be added to the set without violating the disjointness condition, $\boldsymbol{E}$ is maximum in the sense of inclusion: $\nexists$ an alignment path $P' \in G(M,a)$ such that $S(P') \geqslant \alpha$ and such that $\forall P \in \boldsymbol{E}$, $P \| P'$.

Property 18. *Condition* (3) *implies that a solution set contains at least a longest alignment path*: $\exists \hat{P} \in \boldsymbol{E}$ *such that* $\forall$ *alignment path* P *in* $G(M,a)$, $L(\hat{P}) \geqslant L(P)$.

4.2. A disjointness definition

The disjointness definition used in the new search method declares a range of match positions, including intervening insert positions, as a *profile protected region.* Usually, the profile protected region covers positions having a high degree of biological significance, the remaining positions are mainly used to refine the motif description. To be *disjoint*, two profile alignment paths must have two disjoint *sequence protected regions.* This definition is illustrated in Fig. 3. The disjointness definition introduced in [18] for pairwise alignments, which only requires that two alignments do not share one common match operation, is less suitable in a motif search situation.

Definition 19. To a profile of size m, we can assign a *profile protected region* defined by the range $[m_1, m_2]$ such that $0 \leqslant m_1 \leqslant m_2 \leqslant m$.

Definition 20. Given a profile M with a protected region $[m_1, m_2]$, a sequence a and a path P in $G(M,a)$, we define the *sequence protected region* of P by the range

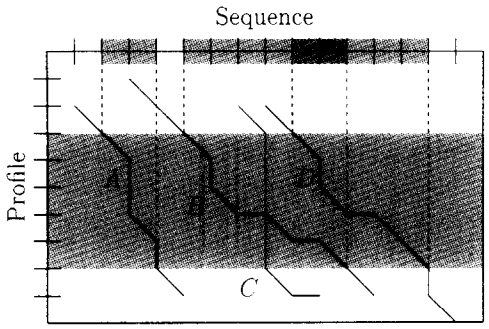

Fig. 3. Only alignment paths B and D are not disjoint. As it has no sequence protected region, alignment path C is disjoint from any alignment path.

$[n_1, n_2]$ where $\mathscr{V}_{x_1,n_1}$ and $\mathscr{W}_{x_2,n_2}$ are respectively the first and the last vertices $\mathscr{X}_{x,y}$ of P satisfying one of the following conditions:

- $\mathscr{X} = \mathscr{M}$ and $m_1 \leqslant x \leqslant m_2$;
- $\mathscr{X} = \mathscr{I}$ and $m_1 \leqslant x < m_2$.

We define $y_{\text{in}}(P) = n_1 - 1$ and $y_{\text{out}}(P) = n_2$ as the *in-index* and the *out-index* of P.

Definition 21. Given a sequence a and a profile M with a protected region $[m_1, m_2]$, two paths in $G(M,a)$ are *disjoint* if and only if their sequence protected regions are disjoint. As this definition is applicable to pairs of alignment paths in $G(M,a)$, it is applicable to pairs of non empty alignments between a and M.

In the presented motif search algorithm, we suppose that any alignment path whose length is greater than or equal to the cut-off value has a sequence protected region. If an alignment P with $L(P) \geqslant \alpha$ has no sequence protected region (see alignment path C in Fig. 3), the solution of the stated problem would have an exponential size, as very many alignment paths surrounding P would also belong to a solution set.

5. A new motif search algorithm

The presented motif search algorithm is based on the dynamic programming method used to find a best alignment between the profile and the sequence. Using a well known dynamic programming method yields an efficient and exact algorithm to solve the stated search problem. In what follows, we solve the motif search problem given a profile M of size m with a protected region $[m_1, m_2]$, a sequence a of size n, a cut-off value α and the disjoint relationship defined in Definition 21.

5.1. A first algorithmic approach

The simplest way to solve the stated search problem consists in repeating iteratively the best alignment algorithm, avoiding to scan the sequence protected regions of the alignments in the current solution set. The resulting algorithm complexity is $O(mnq)$ where q is the number of alignments appearing in the final solution set. The main idea of the new motif search technique is based on the reduction of the sequence scan, which can be accomplished doing two things:

(1) avoiding to scan again already visited uninteresting regions of the sequence;
(2) putting an alignment in the solution as soon as possible (in particular before reaching the end of the sequence).

5.2. New algorithmic approach

The new algorithm scans the sequence, keeping track of the alignment path which could be added next to the solution set. When some conditions are satisfied, this alignment path is effectively put in the solution set, and the alignment moves back in the sequence. In order to describe the algorithm and to show its adequacy to the motif search problem, we define the *algorithm state* as follows.

Definition 22. Observing the algorithm execution, the *algorithm state* is defined by the pair $(\hat{E}, \hat{y})$, where $\hat{E}$ is the current solution set and $\hat{y}$, the current position in the sequence. At the beginning of the algorithm, $(\hat{E}, \hat{y}) = (\emptyset, 1)$.

Definition 23. Given an algorithm state $(\hat{E}, \hat{y})$, the *candidate alignment path* $\hat{P}(\hat{E}, \hat{y})$ is a longest alignment path in $G(M, a)$ disjoint from any alignment path in $\hat{E}$ whose out-index is less than or equal to $\hat{y}$.

Fig. 4 gives a sample of an algorithm state with its corresponding candidate alignment path.

The structure of the new search algorithm is the following:

```
MULTI-ALIGN (profile M, sequence a)
(Ê, ŷ) = (∅, 1);
repeat
    compute P̂(Ê, ŷ) (Section 5.3);
    if acceptance conditions satisfied (Section 5.4) then
      Ê = Ê ∪ P̂(Ê, ŷ);
      compute ŷ to move back in the sequence (Section 5.5);
    else
      compute ŷ to progress in the sequence (Section 5.5);
until end of sequence reached;
return Ê;
```

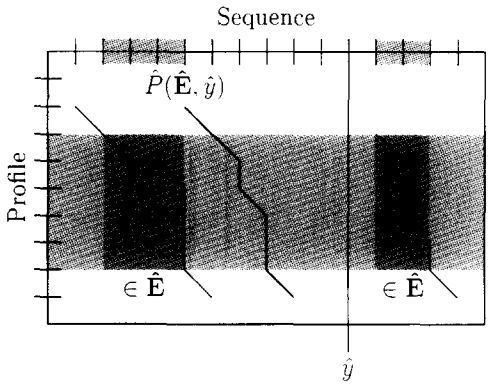

Fig. 4. Sample of an algorithm state $(\hat{E}, \hat{y})$ with its candidate alignment path $\hat{P}(\hat{E}, \hat{y})$.

5.3. *Computation of the candidate alignment path* $\hat{P}(\hat{E}, \hat{y})$

Given an algorithm state $(\hat{E}, \hat{y})$, we denote by $\tilde{P}(\hat{E}, \hat{y})$ a longest alignment path in $G(M, a)$ disjoint from any alignment path in $\hat{E}$ and such that its out-index is equal to $\hat{y}$. This path allows to compute $\hat{P}(\hat{E}, \hat{y})$ when the algorithm comes from state $(\hat{E}^{-}, \hat{y}^{-})$ in the following cases:

- *the algorithm starts or has moved back in the sequence*; the new candidate alignment path $\hat{P}(\hat{E}, \hat{y})$ is simply set to $\tilde{P}(\hat{E}, \hat{y})$;
- *the algorithm has moved forward in the sequence*; if $L(\tilde{P}(\hat{E}, \hat{y})) > L(\hat{P}(\hat{E}^{-} = \hat{E}, \hat{y}^{-}))$ then the new candidate path $\hat{P}(\hat{E}, \hat{y})$ is updated by $\tilde{P}(\hat{E}, \hat{y})$, otherwise $\hat{P}(\hat{E}, \hat{y}) = \hat{P}(\hat{E}^{-}, \hat{y}^{-})$.

5.3.1. *Computation of* $\tilde{P}(\hat{E}, \hat{y})$

Given a current solution set $\hat{E}$, we denote by $\hat{P}_{\mathscr{B}}(\hat{E}, \mathscr{V})$ a longest path in $G(M, a)$ from vertex $\mathscr{B}^{+}$ to vertex $\mathscr{V}$ disjoint from any alignment path in $\hat{E}$. Path $\hat{P}_{\mathscr{E}}(\hat{E}, \mathscr{V})$ is defined in the same manner but for vertices $\mathscr{V}$ and $\mathscr{E}$. Paths $\hat{P}_{\mathscr{B}}(\hat{E}, \mathscr{V})$ and $\hat{P}_{\mathscr{E}}(\hat{E}, \mathscr{V})$ are computed using a dynamic programming technique analogous to the one described in Section 3.3.1. In order to describe the computation of $\tilde{P}(\hat{E}, \hat{y})$, we introduce a path concatenation operator.

Definition 24. Given two paths

$$P_1 = \{V_1V_2, \ldots, V_{r-1}V_r\} \quad \text{and} \quad P_2 = \{W_1W_2, \ldots, W_{s-1}W_s\}$$

such that $V_r = W_1$, we define the path concatenation operator $\circ$ by

$$P_1 \circ P_2 = \{V_1V_2, \ldots, V_{r-1}V_r, W_1W_2, \ldots, W_{s-1}W_s\}.$$

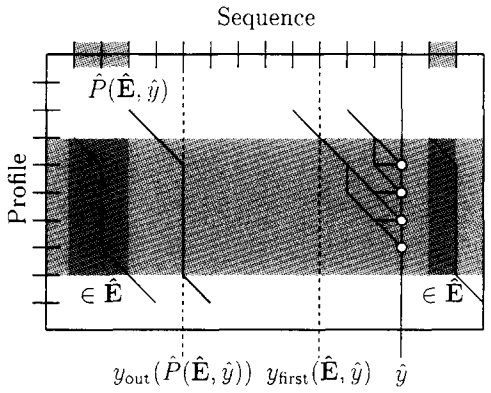

Fig. 5. $y_{\text{out}}(\hat{P}(\hat{E}, \hat{y})) \leqslant y_{\text{first}}(\hat{E}, \hat{y})$: alignment path $\hat{P}(\hat{E}, \hat{y})$ is put is the solution set.

As the out-index of $\tilde{P}(\hat{E}, \hat{y})$ is equal to $\hat{y}$, we choose the longest path among the following one:

- $\hat{P}_{\mathscr{B}}(\hat{E}, \mathscr{V}_{x,\hat{y}}) \circ \{\mathscr{V}_{x,\hat{y}}\mathscr{E}^{+}_{x,\hat{y}}, \mathscr{E}^{+}_{x,\hat{y}}\mathscr{E}\}$ with $m_1 \leqslant x < m_2$ and $\mathscr{V} \in \{\mathscr{M}, \mathscr{I}\}$;
- $\hat{P}_{\mathscr{B}}(\hat{E}, \mathscr{D}_{x,\hat{y}}) \circ \{\mathscr{D}_{x,\hat{y}}\mathscr{E}^{+}_{x,\hat{y}}, \mathscr{E}^{+}_{x,\hat{y}}\mathscr{E}\}$ with $m_1 < x < m_2$;
- $\hat{P}_{\mathscr{B}}(\hat{E}, \mathscr{V}_{m_2,\hat{y}}) \circ \hat{P}_{\mathscr{E}}(\hat{E}, \mathscr{V}_{m_2,\hat{y}})$ with $\mathscr{V} \in \{\mathscr{M}, \mathscr{D}\}$.

5.4. The acceptance condition

A clever set of acceptance conditions is the key to the algorithm efficiency, the sooner the alignment paths are put in the solution set, the less the sequence is scanned.

Definition 25. Given an algorithm state $(\hat{E}, \hat{y})$, the *first-index* $y_{\text{first}}(\hat{E}, \hat{y})$ denotes the lowest in-index of the paths $\hat{P}_{\mathscr{B}}(\hat{E}, \mathscr{V}_{x,\hat{y}})$ such that $\mathscr{V} \in \{\mathscr{M}, \mathscr{I}\}$ and

$$x \in \{m_1, \ldots, m_2 - 1\}.$$

The first-index is determined during the computation of $\tilde{P}(\hat{E}, \hat{y})$. The candidate alignment path $\hat{P}(\hat{E}, \hat{y})$ is put in the solution set $\hat{E}$ when its length is greater than or equal to α and when one of the following conditions is satisfied:

(1) the algorithm has reached the end of the sequence, $\hat{y} = n$;
(2) the algorithm meets a sequence protected region, which means that $\hat{y} + 1$ belongs to the sequence protected region of an alignment path in $\hat{E}$;
(3) any future candidate alignment further in the sequence is disjoint from $\hat{P}(\hat{E}, \hat{y})$, which means that $y_{\text{out}}(\hat{P}(\hat{E}, \hat{y})) \leqslant y_{\text{first}}(\hat{E}, \hat{y})$; Fig. 5 illustrates this condition.

5.5. Update of the algorithm state

Suppose the algorithm moves from state $(\hat{E}, \hat{y})$ to state $(\hat{E}^{+}, \hat{y}^{+})$. There are two kinds of state transitions:

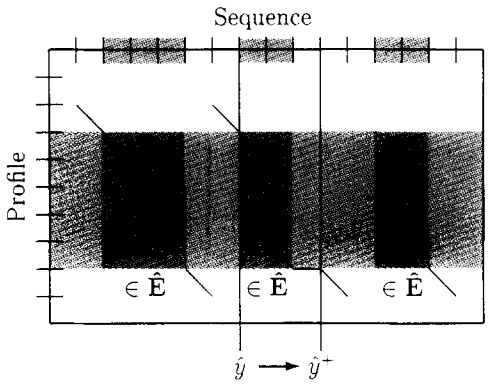

Fig. 6. Jump over a sequence protected region.

- if conditions to put $\hat{P}(\hat{E}, \hat{y})$ in the solution set are not satisfied, then $\hat{E}^+ = \hat{E}$; the algorithm moves forward in the sequence;
- if conditions to put $\hat{P}(\hat{E}, \hat{y})$ in the solution set are satisfied, then $\hat{E}^+ = \hat{E} \cup \{\hat{P}(\hat{E}, \hat{y})\}$; the algorithm moves back in the sequence.

In the first case, no alignment can be put in the solution set, then $\hat{E}^+ = \hat{E}$ and the algorithm simply moves forward in the sequence jumping over the sequence protected regions. The new index $\hat{y}^+$ is set to the lowest index $y \in]\hat{y}, n]$ such that y does not belong to a sequence protected region of an alignment path in $\hat{E}^+ = \hat{E}$ (see Fig. 6). If such an index does not exist, the algorithm stops.

In the second case, instead of returning to the beginning of the sequence, the algorithm maintains a *back-index*.

Definition 26. Given an algorithm state $(\hat{E}, \hat{y})$, the *back-index* $y_{\text{back}}(\hat{E}, \hat{y})$ is the lowest index $y \leqslant \hat{y}$ such that there exists an alignment path P disjoint from any alignment path in $\hat{E}$ with $y_{\text{out}}(P) = y$ and $L(P) \geqslant \alpha$.

Note 2. *Before the algorithm moves back in the sequence, the length of the candidate alignment path is greater than or equal to* α. *Therefore, the back-index always exists in this case.*

The new index $\hat{y}^+$ is set to the lowest index $y \in [y_{\text{back}}(\hat{E}, \hat{y}), n]$ such that y does not belong to a sequence protected region of an alignment path in $\hat{E}^+ = \hat{E} \cup \{\hat{P}(\hat{E}, \hat{y})\}$. As in the first case, if such an index does not exist, the algorithm stops.

5.6. *Adequacy to the motif search problem*

Proposition 27. *The solution set* $\boldsymbol{E}$ *satisfies the minimal length and the disjointness conditions* (1) *and* (2).

Proof. As $\hat{P}(\hat{\boldsymbol{E}}, \hat{y}) \geqslant \alpha$ when it is put in the solution set, the minimal length condition is trivially satisfied. Let P_k denote the kth alignment path put in P by the algorithm. Given any $i, j \in \{1, \ldots, |\boldsymbol{E}|\}$ with $j > i$, $\hat{P}(\hat{\boldsymbol{E}}, \hat{y})$ definition implies that $P_j \parallel P_i$. As the disjointness relationship is symmetric, given any $k, l \in \{1, \ldots, |\boldsymbol{E}|\}$ with $k \neq l$, $P_k \parallel P_l$, therefore the disjointness property (2) is satisfied by $\boldsymbol{E}$. □

Proposition 28. *The solution set* $\boldsymbol{E}$ *satisfies the set maximality condition* 4.

Proof. The algorithm terminates when it reaches the end of the sequence and when $L(\hat{P}(\hat{\boldsymbol{E}}, \hat{y})) < \alpha$. Therefore, no alignment path whose length is greater than or equal to the cut-off value is disjoint from the alignment paths in $\boldsymbol{E} = \hat{\boldsymbol{E}}$. □

Proposition 29. *The solution set* $\boldsymbol{E}$ *satisfies the local optimality condition* (3).

Proof. Suppose that the local optimality condition is not satisfied by the solution set. Then, there exists an alignment path $P \in G(M, a)$ such that for any alignment path $X \in \boldsymbol{E}$ not disjoint from P, $L(X) < L(P)$. Let $\boldsymbol{Q}$ denote the set of alignment paths in $\boldsymbol{E}$ not disjoint from P. We will show that the existence of P implies that there exists an alignment P' with $L(P') \geqslant L(P)$, and such that the set $\boldsymbol{Q}'$ of alignment paths in $\boldsymbol{E}$ not disjoint from P' is strictly included in $\boldsymbol{Q}$. We repeat this implication iteratively, setting $P = P'$ and $\boldsymbol{Q} = \boldsymbol{Q}'$, until $\boldsymbol{Q} = \emptyset$. We finally conclude that there exists an alignment path P^* with $L(P^*) \geqslant L(P) \geqslant \alpha$ such that any alignment path in $\boldsymbol{E}$ is disjoint from P^*, which refutes the set maximality condition.

Let Q denote the first alignment path in $\boldsymbol{Q}$ added to the solution set by the algorithm. From alignment path P, we are going to build an alignment path P' such that $L(P') \geqslant L(P)$, disjoint from any alignment path in $\boldsymbol{E} \setminus \boldsymbol{Q}$ and disjoint from Q. Then, the set $\boldsymbol{Q}'$ of alignment paths in $\boldsymbol{E}$ not disjoint from P' is strictly included in $\boldsymbol{Q}$.

Just before adding Q to the solution set, the algorithm state is $(\hat{\boldsymbol{E}}, \hat{y})$ with $\hat{\boldsymbol{E}} \subset \boldsymbol{E} \setminus \boldsymbol{Q}$ where $Q = \hat{P}(\hat{\boldsymbol{E}}, \hat{y})$. As P is disjoint from any alignment path in $\hat{\boldsymbol{E}}$ and $L(P) > L(Q)$, the definition of the candidate alignment path implies that the out-index of P is greater than $\hat{y}$. This inequality implies that $\hat{y} < n$ and, as P is disjoint from any $X \in \hat{\boldsymbol{E}}$, that index $\hat{y} + 1$ does not belong to a sequence protected region of an alignment in $\hat{\boldsymbol{E}}$. Therefore, Q is added to the solution set thanks to the acceptance condition (3). Fig. 7 shows the algorithm state $(\hat{\boldsymbol{E}}, \hat{y})$.

As $y_{\text{out}}(P) > \hat{y}$ and $y_{\text{in}}(P) < y_{\text{out}}(Q) \leqslant \hat{y}$, there exists a vertex $\mathscr{V}_{x,\hat{y}}$ in P such that $\mathscr{V} \in \{\mathscr{M}, \mathscr{I}\}$ and $m_1 \leqslant x < m_2$ ($\mathscr{M}_{x,\hat{y}}$ in Fig. 7). We can express P as the concatenation of two paths $\cdot P_1 \circ P_2$, where $\mathscr{V}_{x,\hat{y}}$ is the last and the first vertex of P_1 and P_2 respectively. We build a new alignment path defined by $P' = \hat{P}_{\mathscr{B}}(\hat{\boldsymbol{E}}, \mathscr{V}_{x,\hat{y}}) \circ P_2$ as shown in Fig. 8. As the acceptance condition implies that the out-index of Q is lower than or equal to the in-index of P', Q and P' are disjoint. As $y_{\text{in}}(P') > y_{\text{in}}(P)$ and $y_{\text{out}}(P') = y_{\text{out}}(P)$, P' is disjoint from any alignment path in $\boldsymbol{E} \setminus \boldsymbol{Q}$. On the other hand, the construction of P' implies that $L(P') \geqslant L(P)$. □

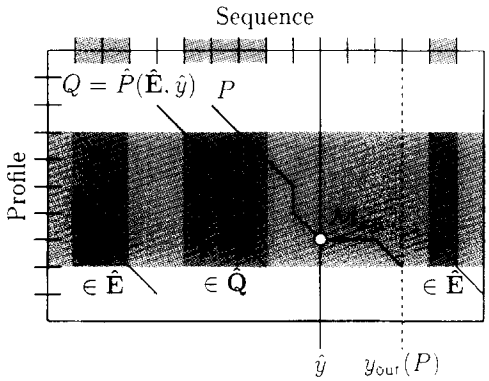

Fig. 7. Algorithm state just before adding Q to the solution set. We note that $y_{\text{out}}(P)$ is greater than $\hat{y}$.

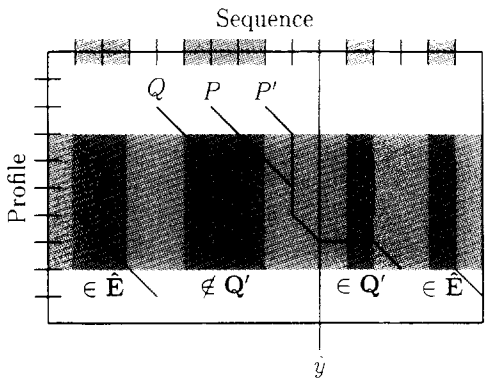

Fig. 8. The new path P' has a length greater than or equal to the length of P. Moreover, P' is disjoint from Q and from any alignment path in $\boldsymbol{E} \setminus \boldsymbol{Q}$, then the set $\boldsymbol{Q}'$ is strictly included in $\boldsymbol{Q}$.

6. Results and conclusion

6.1. Motif search examples

To illustrate some motif search results, the following are three typical motif search applications.

(1) Fig. 9 shows the search result of the SH2 domain in the human GAP protein.[1] The algorithm found two motif occurrences. The profile protected region covers nearly the complete profile because we are not interested in finding overlapping motif occurrences.

[1] Swiss-Prot AC/ID: P20936/GTPA_HUMAN.

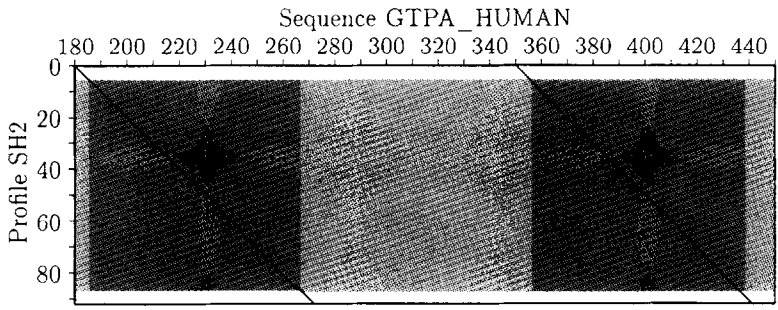

Fig. 9. Search of SH2 domain in protein GTPA_HUMAN.

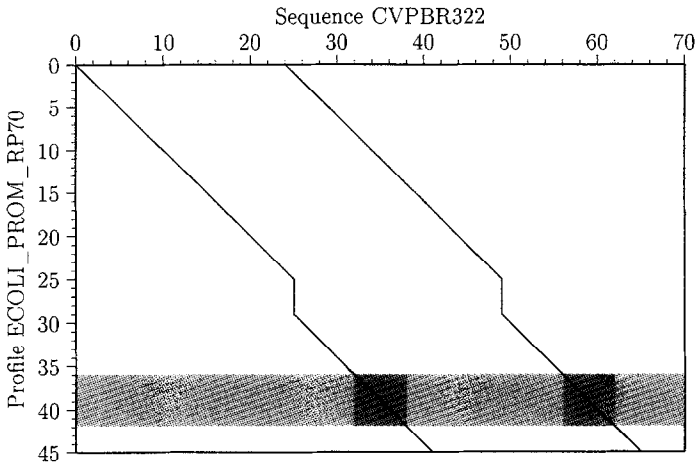

Fig. 10. Search of *E.coli* promoter in DNA sequence CVPBR322.

(2) Fig. 10 shows the search result of the well known Ecoli promoter in a 70bp fragment of DNA sequence pBR322.[2] The algorithm found two occurrences with an important overlap. Indeed, the profile protected region is very small and corresponds to the kernel of the promoter description.

(3) Fig. 11 shows the search result of the SH3 domain in the human PI3-kinase p85 α-subunit protein.[3] In this case, the algorithm finds two occurrences of the motif which represents two alternative alignments in the same region of the sequence. Such results are possible specifying small profile protected regions.

6.2. *Performances*

As a performance measure, we consider the number of sequence symbols scanned by the algorithm. Given a sequence of length n, the first proposed algorithm in Sec-

[2] Swiss-Prot AC/ID: J01749/CVPBR322.
[3] Swiss-Prot AC/ID: P27986/P85A_HUMAN.

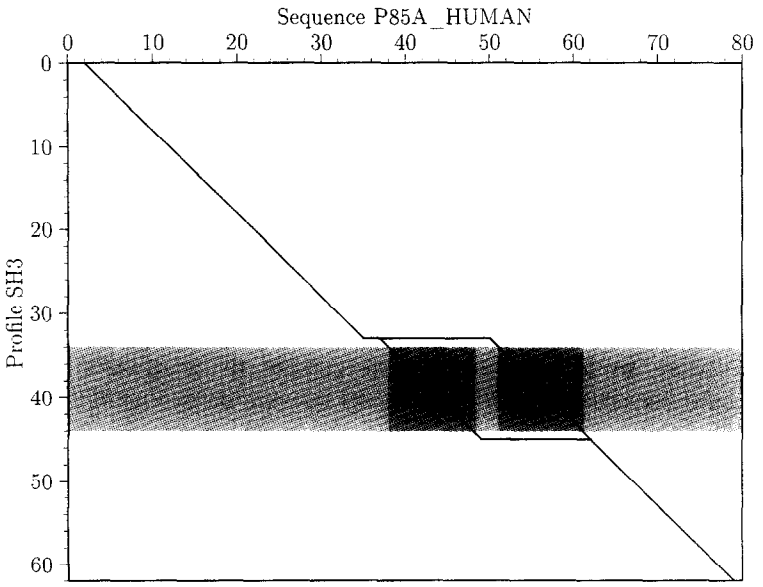

Fig. 11. Two alternative alignments during the search of the SH3 domain in the protein P85A_HUMAN.

Table 3
Comparison between the number of scanned symbols by the new algorithm and by the old algorithm for the SH3 domain search

Sequence Swiss-Prot AC/ID	n	q	N	M'	N'
P34092/MYSB_DICDI	1111	1	1114	51	2171
P14598/NCF1_HUMAN	390	2	568	100	1070
P16333/NCK_HUMAN	377	3	676	149	1508
P23727/P85A_BOVIN	724	1	854	67	1381
P16885/PIP5_HUMAN	1252	1	1427	51	2453
P40996/SCD2_SCHPO	536	2	754	106	1502
P29355/SEM5_CAEEL	228	2	387	101	583
P00523/SRC_CHICK	532	1	736	52	1012
P26674/STE6_SCHPO	911	1	1026	51	1771
P15498/VAV_HUMAN	846	2	968	90	2448
P43603/YFJ4_YEAST	373	1	376	52	694
P38822/YHR4_YEAST	633	2	705	104	1795
Q07157/ZO1_HUMAN	1736	1	1838	59	3413

tion 5.1, must scan $(q+1)\cdot n-M'$ symbols, where q is the number of found alignments, and M' the number of protected symbols in the final solution. For the SH3 domain search in different protein, Table 3 shows the number of scanned symbols by the new search method N, and by the first proposed algorithm N'.

We notice that the new search algorithm becomes more and more efficient when the number of motif occurrences increases. We can express the number of scanned symbols by the new algorithm as $N = n + M$ where M is the number of scanned symbol due

to the moves back in the sequence, therefore M mainly depends on the number of found occurrences and on the number of protected profile positions. Table 3 shows that M remains clearly smaller than the sequence length n. A spectacular result can be illustrated by the search of the Ecoli promoter, defined by a profile with a small protected region, in the DNA sequence CVPBR323. The sequence length is $n=4361$, and with a number of motif occurrences of $q=35$, and $M'=108$ protected symbols, the new algorithm scans only $N=4451$ symbols against $N'=313884$ for the old algorithm.

6.3. Conclusion

In this paper, we have presented a new concept, the generalized profile, to describe biomolecular sequence motifs. This concept is based on two notions, namely the alignment notion, which helps to understand sequence degeneracy, and the score notion, which quantifies the preservation of a motif in a sequence. Associating to generalized profiles a cut-off value and a definition of disjointness between alignments, we have stated a formal motif search problem which answers to a set of typical biological questions. Finally, a new efficient motif search algorithm has been presented, and it has been shown that it solves the stated motif search problem exactly. Results given in Section 6.2 show that this algorithm is barely less efficient than the algorithm for finding a best alignment. This implies that this algorithm can be used for a motif search application involving the complete database of currently known sequences, and thus can be applied to whole genome functional annotation. In spite of the efficiency of the presented algorithm, the main component determining the quality of a motif search remains the profile itself. If the profile is badly defined, the motif search result makes only little sense.

7. For further reading

The following Refs. [1–4, 9, 11–14, 17] are also of interest to the reader.

References

[1] S.F. Altschul, Amino acid substitution matrices from an information theoric perspective, JMB 219 (1991) 555–565.

[2] A. Bairoch, The prosite dictionary of sites and patterns in proteins, its current status, NAR 21 (1993) 3097–3103.

[3] G.J. Barton, M.J. Sternberg, Flexible protein sequence patterns. A sensitive method to detect weak structural similarities, JMB 212 (1990) 389–402.

[4] P. Bucher, A. Bairoch, A generalized profile syntax for biomolecular sequences motifs and its function in automatic sequence interpretation, in: R. Altman, D. Brutlag, P. Karp, R. Lathrop, D. Searls (Eds.), Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology, Menlo Park, AAAI Press, New York, 1994, pp. 53–61.

[5] P. Bucher, K. Karplus, N. Moeri, K. Hofmann, A flexible motif search technique based on generalized profiles, Comput. Chem 20 (1995) 3–23.

[6] O. Gotoh, An improved algorithm for matching biological sequences, JMB 162 (1982) 705–705.
[7] M. Gribskov, R. Luethy, D. Eisenberg, Profile analysis, Meth. Enzymol 183 (1990) 146–159.
[8] M. Gribskov, M. McLachlan, D. Eisenberg, Profile analysis: detection of distantly related proteins, PNAS 84 (1987) 4355–4358.
[9] X. Huang, W. Miller, Comparison of bio-sequences, Adv. Appl. Math. 12 (1991) 337–357.
[10] A. Krogh, M. Brown, I.S. Mian, K. Sjolander, D. Haussler, Hidden Markov Models in Computational Biology. Applications to Protein Modeling, JMB 235 (1994) 1501–1531.
[11] E.S. Lander, M.S. Waterman, Calculating the Secrets of Life, National Academy Press, Washington, DC, 1995.
[12] R. Luethy, I. Xenarios, J.P. Bucher, Improving the sensitivity of the sequence profile method, Protein Sci. 3 (1994) 139–146.
[13] S.B. Needleman, C.D. Wunsch, A general method applicable to the search for similarities in the amino acid sequence of two proteins, JMB 48 (1970) 448–453.
[14] W.R. Pearson, W. Miller, Dynamic programming algorithms for biological sequence comparison, Methods Enzymol 210 (1992) 575–601.
[15] D. Sankoff, J.B. Kruskal, Time Warps, String Edits, and Macromolecules: the Theory and Practice of Sequence Comparison, publisher Addison-Wesley, Reading, MA, 1983.
[16] T.F. Smith, M.S. Waterman, Identification of common molecular subsequences, JMB 147 (1981) 195–197.
[17] M.S. Waterman, Introduction to Computational Biology, Chapman & Hall, New York, 1995.
[18] M.S. Waterman, M. Eggert, A New algorithm for Best Subsequence Alignments with Application to tRNA-rRNA Comparisons, JMB 197 (1987) 723–728.